

REMARKS

In the Office Action mailed June 6, 2008, Claims 4-5 and 29-30 were objected to as reciting informalities. Claims 1-2, 4-10, and 12-50 were rejected under 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Pat. Pub. No. 2007/0233871 ("*Fletcher*"). Applicants respectfully traverse the objections and rejections.

By this amendment, Claims 1, 5-6, 17, 20-24, 28, 30-32, and 49-50 are amended to more fully clarify the claimed invention. Applicants respectfully request reconsideration of the pending application in light of the amendments and the remarks below. Each issue raised in the Office Action is addressed hereinafter.

THE CLAIM OBJECTIONS

Claims 4-5 and 29-30 were objected to because Claim 4 appears identical to Claim 5 and Claim 29 appears identical to Claim 30. Claims 4 and 29 are canceled by this amendment. Additionally, Claims 12 and 45, which depend from Claims 4 and 29 respectively, are also canceled by this amendment. Removal of the objection is respectfully requested.

THE § 102 REJECTIONS

The Office Action contends that *Fletcher* anticipates Claim 1. Of course, a claim is anticipated only if each and every element of the claim is found in a single prior art reference. Further, the identical invention must be shown in as complete detail as shown in the claim. MPEP § 2131. Since all features of Claim 1 are not found in *Fletcher*, removal of the rejection of Claim 1 is respectfully requested.

Present Claim 1 recites:

A method for handling requests for web services, the method comprising the computer-implemented steps of:
receiving at a web services broker, from a particular instance of a client application, a request for information, wherein said request includes an identification of a particular web service from which said particular instance wants said requested information, the request having first input data, the first input data being in a form that cannot be used by said particular web service to service requests for said information;
wherein the particular web service serves as the source of said requested information, and is separate from the web services broker;
wherein the particular instance of said client application is separate from the web services broker and does not have logic for directly interacting with said particular web service;
in response to receiving said request, the web services broker
accessing, based on said identification of said particular web service, transformation information that specifies,
 how to transform said first input data associated with said request to second input data that said particular web service can use to service requests for said requested information, and
 how to invoke said particular web service in a manner required by said particular web service, to obtain said requested information from said particular web service;
transforming said first input data to said second input data; and
invoking, in said manner required by said particular web service, said particular web service to obtain said requested information from said particular web service.

As a preliminary matter, Applicants note that Claim 1 is not about general transformation of data passed between components of a web services architecture. Rather, Claim 1 specifically features, among other things, brokering a **request** from a **client application** for a particular web service identified in the request. As part of brokering the request from the client application, Claim 1 features accessing transformation information based on the particular web service identified in the request and then, among other steps, transforming input data specified in the request into a form that can be used by the particular web service identified in the request. The approach of Claim 1 for brokering a request from a client application enables, for example, development of a generic client application that is configured to request information from a

particular web service without having to be configured to provide input data in a form that the web service can use and without having to be configured to invoke the particular web service in a manner required by the web service.

In rejecting Claim 1, the Office Action does not explicitly identify what element of *Fletcher* it equates with the "client application" of Claim 1. From the content of *Fletcher* and the rejection of Claim 1, Applicants have identified two possible elements of *Fletcher* that the Office Action might mean to equate with the "client application" of Claim 1. Specifically, it appears that the Office Action might mean to equate the "client application" of Claim 1 with *Fletcher's* "application or software resource requesting a particular service" (hereinafter referred to as "*Fletcher's* application") or the Office Action might mean to equate the "client application" of Claim 1 with *Fletcher's* "portlet" which is described in *Fletcher* as "an intermediary between an application or software resource requesting a particular service and a software resource providing that service." (*Fletcher*, para. 46). In either case, as shown below, whether the "client application" of Claim 1 is equated with *Fletcher's* application or *Fletcher's* portlet, each and every feature of Claim 1 is not satisfied by *Fletcher*.

Preliminarily, however, it is critical to understand the differences between Claim 1 and *Fletcher*. Specifically, to understand how the portal platform of *Fletcher* processes data passed between web service intermediaries differently from how the portal platform processes data it receives in requests from client applications and that it forwards to software resources that are separate from the portal platform and that serve as the source of requested information.

DIFFERENCE BETWEEN *FLETCHER'S* "CONVERT-LETS" AND *FLETCHER'S* "PLUG-LINK MECHANISM"

Fletcher describes a portal platform in which "portlets" are used to aggregate external web services into the portal platform. (*Fletcher*, para. 87). A portlet, as described in *Fletcher*, is a "web service intermediar[y]" or "web service prox[y]" that provides a "standard interface for a web service aggregated in the portal platform." (*Fletcher*, para. 46 and 87). As shown in Figure 18 and described in paragraph 85-86 of *Fletcher*, a "plug-link" mechanism is used by the portal platform to map (not convert or transform) a portlet's standard interface to the interface of an aggregated web service. *Fletcher* describes the "plug-link" mechanism as providing a "relatively restricted form of data mapping" that is "limited to specifying a source message, a single target message, a source message attribute, and a single target message attribute." (*Fletcher*, para. 86). The restricted form of data mapping provided by the plug-link mechanism is used by the portal platform of *Fletcher* to pass data between a portlet and a separate web service that serves as a source of information (see e.g., *Fletcher*, Figure 18, elements 1820, 1830, and 1840).

In contrast, to transform data passed between portlets, the portal platform employs a convert-let. A "convert-let" is a "**portlet-to-portlet** operation transformation." (Emphasis added) (*Fletcher*, para. 87). However, Applicants note that by passing data between portlets through a covert-let, a portlet does not request another portlet to perform a service. In other words, portlets are not clients of each other. Further, portlets are not web services in the sense that they serve as the source of requested information. Rather, portlets are "web service intermediaries" or "web service proxies" for software resources that serve as the source of requested information. (*Fletcher*, para. 46).

Thus, in *Fletcher*, plug-links are used to map (not transform or convert) a single parameter in a message provided to a portlet's standard interface to a single parameter in a

message provided to the web service for which the portlet is acting as a web service intermediary or proxy. Covert-lets, on the other hand, are used to transform **only** data passed between portlets.

Fletcher's application and *Fletcher's* portlet, separately equated with the "client application" of Claim 1, will each now be analyzed in light of the differences identified above between *Fletcher's* plug-links and *Fletcher's* covert-lets.

FLETCHER'S APPLICATION AS THE CLAIMED "CLIENT APPLICATION" DOES NOT
SATISFY ALL FEATURES OF CLAIM 1

Even assuming that the Office Action means to equate *Fletcher's* application with the claimed "client application," each and every limitation of Claim 1 is not satisfied by *Fletcher*. For example, Claim 1 features a "request" from "a particular instance of a client application" that "includes **an identification of a particular web service** from which said particular instance wants said requested information." While *Fletcher* does state that a portlet may act as an intermediary "between an application requesting a **particular service** and a software resource providing that **service**" (*Fletcher*, para. 46) there is nothing in *Fletcher* that suggests that requests from applications include "an identification of a particular web service from which said particular instance wants said requested information" as featured in Claim 1. Rather, requests from applications in *Fletcher* request only that they want a particular service, but do not indicate who they want to receive that service from. Therefore, it is respectfully submitted that requests from *Fletcher's* application do not include "an identification of a particular web service from which said particular instance wants said requested information" as featured in Claim 1.

Further, since *Fletcher* does not teach or suggest a "request" from "a particular instance of a client application" that "includes **an identification of a particular web service** from which

said particular instance wants said requested information" as featured in Claim 1, *Fletcher* cannot possibly teach or suggest the following steps of Claim 1 performed by the claimed "web services broker" in response to receiving such a request:

accessing, based on said identification of said particular web service, transformation information that specifies,
how to transform said first input data associated with said request to second input data that said particular web service can use to service requests for said requested information, and
how to invoke said particular web service in a manner required by said particular web service, to obtain said requested information from said particular web service;
transforming said first input data to said second input data; and
invoking, in said manner required by said particular web service, said particular web service to obtain said requested information from said particular web service.

Additionally, as explained above, the plug-link mechanism of *Fletcher* maps only a single source message attribute to a single target message attribute and does not provide the ability to transform data from one form to another. Thus, *Fletcher's* plug-links, unlike the claimed "transformation information," do not specify "how to transform said first input data associated with said request to second input data that said particular web service can use to service requests for said requested information" as featured in Claim 1. Further, the convert-lets of *Fletcher* provide only the ability to transform data passed between portlets and not data passed between a client application and a particular web service identified in a request from the client application.

Therefore, even when *Fletcher's* application is equated with the claimed "client application" the rejection of Claim 1 cannot be maintained because each and every feature of Claim 1 is not satisfied by *Fletcher* as required of a rejection under § 102.

***FLETCHER'S* PORTLET AS THE CLAIMED "CLIENT APPLICATION" DOES NOT
SATISFY ALL FEATURES OF CLAIM 1**

Similarly, the rejection of Claim 1 under § 102 cannot be maintained even assuming the Office Action means to equate *Fletcher's* portlet with the "client application" of Claim 1. As explained previously, *Fletcher's* convert-lets transform data passed between portlets. However, a portlet, when providing data to a convert-let, does not **request** information from a particular web service. Thus, because *Fletcher's* portlets, when providing data to a convert-let, do not request information from a web service they cannot be the claimed "client application" which requests information from a particular web service. In any event, there is nothing in *Fletcher* that suggests that the data passed between portlets through a convert-let includes an identification of a particular software resource. In contrast, Claim 1 expressly features a "request" from a "client application" that includes "an identification of a particular web service from which said particular instance wants said requested information." Thus, the portlet cannot be the "client application" of Claim 1 because it does not make the "request" of Claim 1, or any request for that matter, when providing data to another portlet through a convert-let.

Therefore, even when *Fletcher's* portlet is equated with the claimed "client application" the rejection of Claim 1 cannot be maintained because each and every feature of Claim 1 is not satisfied by *Fletcher* as required of a rejection under § 102.

CLAIM 1 AND CLAIM 17 ARE PATENTABLE OVER *FLETCHER*

Based on the foregoing, Applicants respectfully request withdrawal of the rejection of Claim 1 because each and every feature of Claim 1 is not satisfied by *Fletcher* either in the case when *Fletcher's* application is equated with the claimed "client application" or in the case when *Fletcher's* portlet is equated with the claimed "client application."

Claim 49 recites features similar to those recited by Claim 1 which are also not satisfied by *Fletcher*. Therefore, rejection of Claim 49 should also be withdrawn for the same reasons given above with respect to Claim 1.

CLAIM 17 RECITES DIFFERENT FEATURES THAN CLAIM 1 THAT ARE ALSO NOT
SATISFIED BY *FLETCHER*

The Office Action rejects Claim 17 under the same rationale used to support the rejection of Claim 1. However, Claim 17 recites features not found in Claim 1 that are also not satisfied by *Fletcher*. For example, Claim 17 recites the following bolded features not found in Claim 1 related to a request from a client application that includes **an identification of a particular instance of the client application** and in response to receiving that request, accessing transformation information **based on the identification including in the request:**

...

receiving at a web services broker, from a client application, a request for information, **wherein said request includes an identification of a particular instance of said client application**, the request having first input data, the first input data being in a form that cannot be used by a particular web service to service requests for said information;

...

in response to receiving said request, **based on said identification of said particular instance of said client application**, the web services broker accessing transformation information;

...

The Office Action, by rejecting Claim 17 under the same rationale used to reject Claim 1, has not even alleged that the above bolded limitations are satisfied by *Fletcher*. Further, there is

nothing in *Fletcher* that appears to satisfy these limitations. Thus, the rejection of Claim 17 under § 102 should also be withdrawn.

Claim 50 recites features similar to those recited by Claim 17 which are also not satisfied by *Fletcher*. Therefore, rejection of Claim 50 should also be withdrawn for the same reasons given above with respect to Claim 17.

DEPENDENT CLAIMS

All of the remaining claims depend on one of the claims discussed above. Therefore, each of these claims is allowable for those reasons. In addition, each of the dependent claims separately introduces limitations that render the claims allowable over the art. However, due to the fundamental differences already identified, and to expedite favorable resolution of this case, separate arguments are not provided for these claims.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,

Hickman Palermo Truong & Becker LLP

Date: September 5, 2008

/AdamCStone#60531/
Adam Christopher Stone
Reg. No. 60,531

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080 ext. 231
Facsimile No.: (408) 414-1076